



EMBERI ERŐFORRÁSOK
MINISZTERIUMA



full-stack
fejlesztés



Élő LaTeX dokumentáció készítése Python segítségével

Neumann János Szakkollégium
Óbudai Egyetem
2017. április 11.



NEUMANN JÁNOS SZAKKOLLÉGIUM

Bedők Dávid
2017.04.11.
v0.3



Dokumentáció készítés

- Mérnöki munka dokumentálása
 - Alkalmazás/Termék dokumentálása
 - Project munka dokumentálása
-
- Kik számára készül?
 - Kötelező eleme a munkának, ámbár lényegtelen része?

Dokumentáció fajtái

- **Fejlesztői dokumentáció**
 - Specifikációk, követelményanalízis, futtatási és fejlesztői környezet leírása, kód- és implementációs szabályok (akár WoW is)
 - Konceptió terv
 - Lehetséges megoldások bemutatása és értékelése
 - Rendszerterv
 - Hardware és software környezet, használati esetek, UML
- **Felhasználói dokumentáció**
 - Futtatási környezet, használatai esetek, bementek, kimenetek, mintaesetek, hibaüzenetek jegyzéke
- **Programismertető**
 - Rövid leírás, termék tulajdonságai, minimális hardware és software környezet
- **Telepítési/operátori kézikönyv**
 - Üzembe helyezés részletes leírása

Jó dokumentáció ismérvei

- Ne legyen regény hosszúságú, de ne is hagyja el a lényegi részeket
- Pontosán és világosan tagolt fejezetek
- Tömör
- Olvasható, olvasmányos (?)
- Pontos

Dokumentum formátumok

- UTF8 szövegszerkesztővel szerkeszthető
 - Plain text (*.txt)
 - Wiki text/markup, wikicode
 - HTML (*.x)html)
 - LaTeX (*.tex)
- Tipikusan (saját) alkalmazással szerkeszthető
 - Open Document Format (*.odt)
 - Rich Text Format (*.rtf)
 - WORD (*.doc)
 - Office Open XML (*.docx)
 - Uniform Office Format (*.uof, China)

Népszerű szerkesztő programok

- Bill Joy: vi (1967, BSD)
- Richard Matthew Stallman: Emacs
- The Document Foundation: LibreOffice
- Microsoft: WordPad
- Microsoft: Word ('online' platform is)
- Sun: OpenOffice
- Apple: Pages ('online' platform is)
- Sun: StarOffice
- Corel: WordPerfect
- Google: Docs (kizárólag 'online' platform)

Fontos szempont, hogy platform független szerkesztő programot válasszunk! Az 'online' verziók ennek legtöbbször eleget tesznek.

Plain text vs. binary doc vs. doc xml

- **UTF8 szöveg**
 - Egyszerű, tiszta, legtöbbször egyértelmű
 - Ha formázni szeretnénk, akkor viszont rém álommmá válik ((X)HTML + CSS → dokumentum?)
 - Legtöbbször csak akkor előnyös ha gép generálja valamilyen emberi/gépi forrás(ok)ból (pl. JavaDoc default doclet)
 - Verziókezelés kézenfekvően megoldható
- **Bináris dokumentum**
 - Gazdag formázási lehetőség, nagy múltú cégek, sok tapasztalat, professzionális (?) eszközök
 - Konvertálás problémás, gépi feldolgozás nehézkes és körülményes, legtöbbször nem éri meg a befektetett munkát
 - Verziókezelés csak e célra létrehozott dokumentum kezelőkben oldható meg, melyek körülményesek és nem hatékonyak
- **XML alapú nyílt dokumentum**
 - Gazdag formázási lehetőség, tiszta és nyílt tartalom
 - Áttekinthetősége és egyszerűsége azonban elveszik
 - Gépi feldolgozás már megoldható, de ember számára a nyíltság és áttekinthetőség elméleti síkon marad
 - Verziókövetés megoldható, bár a tömörített elemeket ki kell bontani, ha vannak ⁷

Választás

Milyen célra készül a dokumentum?

Kinek készül a dokumentum?

Milyen gyakran kell frissíteni, karbantartani?

Kell-e verziózni, verziókövetni?

Szükség van-e 'visszavonási' lehetőségre,
history-ra?

Miért LaTeX?

- Mert a dokumentáció lényegi része a tartalom, mely az entrópiát hordozza
 - E tartalmi rész leírására a **plain text** a legalkalmasabb
 - **Verziókövetés** kézenfekvően megoldható (tetszőleges SCM)
- Mert akár a **tartalmi rész** is tetszőleges program vagy script nyelven gyorsan és rugalmasan **legenerálható**
- A dokumentáció **formázása gépiesíthető**, utasításokkal vezérelhető
 - Mert a formázás kevésbé fontos, programozható és különböző témájú dokumentumok között átvihető, kiemelhető (az algoritmus hordozható, függetlenül attól hogy milyen paraméterekkel hívjuk meg)
- Mérnöki szemlélet nélkül is népszerű, ám mérnökök számára a kánaán lehet

Kollaboráció LaTeX-ben

Kézenfekvő, intuitív, grafikus közös munka platformja:

- Google Docs
- DropBox Paper

LaTeX esetében mindez egy GIT repository-val oldható meg

- Az elérhető kollaborációs eszközökhöz képest szaktudást igényel (ez mérnökök esetén nem probléma)
- Merge-ölni kell a változásokat, melyhez bár vannak külső eszközök (pl. kdiff3), messze nem olyan kényelmes mint a valós idejű párhuzamos szerkesztés

Platformfüggetlen LaTeX

Elvben egy UTF8 text editor elég lehet (Notepad++, Emacs, Atom, stb.). De sok apróságot tud hozzáadni egy keretprogram, főleg a fordítás és a plugin-ek kezelése, letöltése vonatkozásában.

<https://www.latex-project.org/get/>

TeX distributions:

- TeX Live (Linux)
- MacTeX (MacOS)
- MiKTeX (Windows)

Szerkesztő:

- **TeXstudio** (<http://www.texstudio.org/>)
(Linux, Mac, Win)

Magyar LaTeX csomag

a', i', o", u", u', stb. → működik, de...

Minden kezdő LaTeX használó szembetalálkozik ezzel.

Részletes leírás és további információ:

<http://math.bme.hu/latex/>

```
..  
\usepackage[latin2]{inputenc}  
\usepackage{lmodern}  
\def\magyarOptions{defaults=hu-min}  
\usepackage[french,magyar]{babel}  
..
```

A magyar TeX állományok “javasolt” kódolása **latin2**, de használható BOM nélküli UTF-8 is. Arra kell elsősorban figyelni, hogy a később egymást felhasználó TeX állományok azonos kódolással készüljenek.

LaTeX előnyös tulajdonságai

- Ezernyi **kereszthivatkozások**. Ne spóroljunk vele!
- **Gyors** munka.
- Text based/content **search** file kezelőkből!
- Bármely SCM-el **verziókezelhető**, követhető.
- Tartalomjegyzék, ábrajegyzék, kódjegyzék. **SOHA** nem kell ezekkel többet foglalkozni.
- **Forráskódok**, program kimenetek külső szöveges állományból beolvashatóak → automatikusan sorszámozott keretbe tehetőek. Vagy rózsaszín pónisba. A kódrészletek nem részei a dokumentumnak, nem zavarják be pl. a keresést, nem kell “átgörgetni” őket, nem veszik el a helyet.
- **Képek** hivatkozásai kerülnek csak be. A képek helyével nem kell foglalkozni. Úgy alapvetően az oldaltörésekkel nem kell foglalkozni. Szinte soha.
- **Szép**.
- Scriptelhető, pluginelhető.
- Programmal könnyen generálható.

Hivatkozások

```
..  
\label{chap:loremipsum}  
..  
\ref{sec:loremipsum}  
..  
\pageref{sec:loremipsum}  
..  
\aref{sec:loremipsum} % MagyarLaTeX!  
..  
\nameref{sec:loremipsum}  
..
```

Bármit meg lehet hivatkozni, visszacapni oldalszámát, nevét, sorszámát.

Tartalomjegyzék

```
..  
\tableofcontents  
..  
\listoffigures  
..  
\lstlistoflistings  
..
```

..ábrák jegyzéke, forráskódok jegyzéke (pl.)

Forráskód beszúrása

```
..  
\lstinputlisting[caption={Lorem  
Ipsum},label={lst:loremipsum}]{source/Lorem  
.java}
```

Lorem.java

```
..  
package hu.qwaevisz.demo;  
  
public interface Lorem {  
  
    int ipsum(String reference);  
  
}
```

A dokumentumszerkesztés a content szerkesztéséről szól, nem a programírásról!

Képek

```
..  
\begin{figure}[h!]  
  \centering\includegraphics[width=100mm]{images/tiger.png}  
  \caption{Sample picture}  
  \label{fig:sample}  
\end{figure}  
..
```

tiger.png



TeX állományok szétválasztása

Akár fejezetenként (nem ajánlás).

viszont ha bizonyos TeX elemeket, pl. táblázatokat dinamikusan szeretnénk létrehozni, kézenfekvő ezen elemeket dinamikusan beemelni a dokumentumba!

original.tex

```
..  
\input{lorem.latin2.tex}  
..
```

lorem.latin2.tex

```
\chapter{Lorem ipsum}  
\label{chap:lorem}  
Lorem ipsum dolor sit amet...
```

Probléma felvetés

Előfordult-e már az, hogy egy dokumentumban szerepeltetni kell olyan elemeket, táblázatokat, melyek teljes egészében megtalálhatóak más helyen (forráskódban, adatbázisban, XML állományban)?

Feladat:

Készítsünk egy magyar nyelvű dokumentumot, melyben megjelenik a dokumentált rendszer által támogatott nyelvek listája.

Környezet

Tegyük fel hogy van egy rendszer, melyhez dokumentációt kell készítenünk a megadott követelmények szerint.

Az elérhető nyelvek listája legyen elérhető egy adatbázis táblában (`language`), melyben minden nyelvhez tartozzon egyedi azonosító és néhány egyéb adat (pl. alapértelmezett dátumformátum).

A rendszer e táblából olvassa ki az elérhető nyelveket, hogy pl. egy lenyíló listából a felhasználó ki tudja választani a számára optimálisat.

A felület természetesen nyelvesített (hiszen mi értelme lenne nyelv választásnak ha nem így lenne). A nyelvek egyedi azonosítója alapján a rendszer kikeresi a nyelvi konstanshoz tartozó feliratot.

Adatbázis

language_grid	language_name	language_constant
10	HUN	L_HUNGARIAN
20	ENG	L_ENGLISH
30	JAP	L_JAPANESE

schema / language.sql

```
CREATE TABLE language (  
  language_unid int(11) NOT NULL AUTO_INCREMENT,  
  language_grid int(11) NOT NULL,  
  language_name varchar(50) NOT NULL,  
  language_constant varchar(50) NOT NULL,  
  language_shortdateformat varchar(50) NOT NULL,  
  language_longdateformat varchar(50) NOT NULL,  
  PRIMARY KEY (language_unid),  
  UNIQUE KEY UI_LANGUAGE_GRID (language_grid)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Initial / language.sql

```
INSERT INTO language  
(language_grid,language_name,language_constant,language_shortdateformat,language_longdateformat) VALUES (10,'HUN','L_HUNGARIAN','Y.m.d.','Y.m.d. H:i:s');  
INSERT INTO language  
(language_grid,language_name,language_constant,language_shortdateformat,language_longdateformat) VALUES (20,'ENG','L_ENGLISH','m.d.Y','m.d.Y H:i:s');  
INSERT INTO language  
(language_grid,language_name,language_constant,language_shortdateformat,language_longdateformat) VALUES (30,'JAP','L_JAPANESE','m.d.Y','m.d.Y H:i:s');
```

Lokalizáció

locale / uk / messages.ini

```
L_HUNGARIAN=Hungarian  
L_ENGLISH=English  
L_SPANISH=Spanish  
L_GERMAN=German  
L_PORTUGUESE=Portuguese  
L_RUSSIAN=Russian  
L_FRENCH=French  
L_JAPANESE=Japanese  
L_SLOVAK=Slovak  
L_ROMANIAN=Romanian  
L_ITALIAN=Italian
```

locale / hu / messages.ini

```
L_HUNGARIAN=Magyar  
L_ENGLISH=Angol  
L_SPANISH=Spanyol  
L_GERMAN=Német  
L_PORTUGUESE=Portugál  
L_RUSSIAN=Orosz  
L_FRENCH=Francia  
L_JAPANESE=Japán  
L_SLOVAK=Szlovák  
L_ROMANIAN=Román  
L_ITALIAN=Olasz
```

Elvárt dokumentációs részlet

Konstans	Megnevezés
ENG	Angol
HUN	Magyar
JAP	Japán

language.latin2.tex

```
\begin{table}[h!]  
  \centering{  
    \begin{tabular}{|c|c|}  
      \hline  
        \textbf{Konstans} & \textbf{Magnevezés} \\ \hline  
        \small\texttt{L\_ENGLISH} & Angol \\ \hline  
        \small\texttt{L\_HUNGARIAN} & Magyar \\ \hline  
        \small\texttt{L\_JAPANESE} & Japán \\ \hline  
      \end{tabular}  
    }  
  \caption{Nyelvek}  
  \label{tab:language_values}  
\end{table}
```

LaTeX dokumentum paraméterek

```
..  
\input{configuration.tex}  
..
```

configuration.tex

```
\newcommand{\SOURCEDIRECTORY}{./outer/source}  
\newcommand{\SCHEMADIRECTORY}{\SOURCEDIRECTORY/database/schema}
```


LaTeX dokumentum

```
..
\chapter{Lokalizáció}
\label{chap:localization}

\lstinputlisting[caption={Nyelv},label={lst:language}]
{\SCHEMADIRECTORY/language.sql}

\lipsum[1]

\input{generated/language.latin2.tex}
..
```



Python

<https://www.python.org/>

Verzió: **2.7.13** vagy **3.6.1**

Magas szintű, általános célú programozási interpretált (script) nyelv, a Java-val kb. egyidős (1991-től beszélhetünk róla).

“Kódolvashatóság” elsődleges célja volt a nyelvnek.

Hibrid nyelvnek tekinthető (objektum-orientált, imperative, funkcionális és procedurális paradigmák is előfordulnak benne).

Nyílt forráskódú és tekinthetjük cross-platformnak (létezik fordító számos OS-re, működési elve a Java-hoz hasonló (virtuális gép, python bytecode).

Python helye a világban

Vitatott kérdés, de ott ahol:

- **script** nyelvre van szükségünk
- fontos a **cross-platform** viselkedés
- elvárt a **gazdag** és sokrétű **osztálykönyvtár** létezése és
- a **magas szintű** programozási környezet

Valamint:

- Nem szeretnénk vagy nem tudunk Java Virtual Machine-t beizzítani

Ott helye lehet a Python nyelvnek.

Terv

Legyen adott a táblázat keretének és egy sorának egy-egy TeX template-je. A Python sscript ezeket beolvasva hozza létre a valós TeX táblázatot.

frame.template

```
\begin{table}[h!]  
  \centering{  
    \begin{tabular}{|c| ##valuewidth##|}  
      \hline  
      \textbf{Konstans} & \textbf{Megnevezés} \\ \hline  
##items##  
    \end{tabular}  
  }  
  \caption{##caption##}  
  \label{tab:##tableid##}  
\end{table}
```

item.template

```
\small\texttt{##constant##} & ##label## \\ \hline
```

Template file-ok beolvasása

create.py

```
import codecs

def loadTemplate( fileName ):
    with codecs.open(fileName, 'r', 'utf-8') as file:
        content=file.read()
    return content
```

```
frame = loadTemplate('frame.template')
item = loadTemplate('item.template')
```

Lokalizációs file beolvasása

create.py

```
def loadLocalizationMessages( fileName ):  
    with open(fileName) as f:  
        lines = f.read().splitlines()  
  
    result = dict()  
  
    for line in lines:  
        pair = line.split('=', 1)  
        result[pair[0]] = unicode(pair[1], 'utf-8')  
  
    return result
```

constants

```
{  
    'L_HUNGARIAN': u'Magyar',  
    'L_SLOVAK': u'Szlov\xek',  
    'L_ENGLISH': u'Angol',  
    'L_JAPANESE': u'Jap\xen',  
    'L_SPANISH': u'Spanyol',  
    'L_RUSSIAN': u'Orosz',  
    'L_ROMANIAN': u'Rom\xen',  
    'L_ITALIAN': u'Olasz',  
    'L_FRENCH': u'Francia',  
    'L_PORTUGUESE': u'Portug\xell',  
    'L_GERMAN': u'N\xemet'  
}
```

```
constants = loadLocalizationMessages(sourcePath + 'locale/hu/messages.ini')
```

Csatlakozás adatbázishoz

create.py

```
import mysql.connector
from mysql.connector import errorcode

def getDbConnection( host, user, password, database ):
    try:
        connection = mysql.connector.connect(user=user, password=password,
        host=host, database=database, charset='utf8')
        return connection
    except mysql.connector.Error as err:
        if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
            print("Something is wrong with your user name or password")
        elif err.errno == errorcode.ER_BAD_DB_ERROR:
            print("Database does not exist")
        else:
            print(err)

connection = getDbConnection('localhost', 'root', 'secret', 'texdb')
```

Natív lekérdezés futtatása és konstans adatokkal való párosítása

create.py

```
def queryLanguages( connection, constants ):
    cursor = connection.cursor()
    query = ("SELECT language_grid AS grid, language_name AS name, language_constant AS
constant FROM language")
    cursor.execute(query)

    result = dict()
    for (grid, name, constant) in cursor:
        result[name] = constants[constant] if constants.has_key(constant) else name

    cursor.close()

    result = collections.OrderedDict(sorted(result.items()))
    return result
```

constants

```
{
    u'HUN': u'Magyar',
    u'JAP': u'Jap\xeln',
    u'ENG': u'Angol'
}
```


Feldolgozás

Template konstansok cseréje

create.py

```
def buildContent( constants, values, tableId, caption, valueWidth ):
    frame = loadTemplate('frame.template')
    item = loadTemplate('item.template')

    items = ''
    for key in values:
        tmp = item
        tmp = tmp.replace('##constant##', key.replace('_', '\_'))
        tmp = tmp.replace('##label##', values[key])
        items += tmp + '\n'

    frame = frame.replace('##valuewidth##', valueWidth)
    frame = frame.replace('##tableid##', tableId)
    frame = frame.replace('##caption##', unicode(caption, 'utf-8') )
    frame = frame.replace('##items##', items)
    return frame
```

```
languages = queryLanguages(connection, constants)
content = buildContent(constants, languages, 'language_values', 'Nyelvek', 'c')
```

Karakterkódolás változtatása

create.py

```
def writeUTF8File( destination, fileName, content ):
    file = codecs.open(destination + fileName, 'w', 'utf-8')
    file.write(content)
    file.close()

def convertFileUTF8toLatin2( destination, sourceFileName, targetFileName ):
    sourceEncoding = 'utf-8'
    targetEncoding = 'iso-8859-2'
    source = open(destination + sourceFileName)
    target = open(destination + targetFileName, 'w')
    target.write(unicode(source.read(), sourceEncoding).encode(targetEncoding))
```

```
writeUTF8File(destination, 'language.utf8.tex', content)
convertFileUTF8toLatin2(destination, 'language.utf8.tex', 'language.latin2.tex')
```

Összesítés

create.py

```
destination = '../generated/'
sourcePath = '../outer/source/'

connection = getDbConnection('localhost', 'root', 'secret', 'texdb')
constants = loadLocalizationMessages(sourcePath + 'locale/hu/messages.ini')

languages = queryLanguages(connection, constants)
content = buildContent(constants, languages, 'language_values', 'Nyelvek', 'c')
writeUTF8File(destination, 'language.utf8.tex', content)
convertFileUTF8toLatin2(destination, 'language.utf8.tex', 'language.latin2.tex')

connection.close()
```

Demo

<https://github.com/davidbedok/oe-latex>

```
> cd \live-latex-demo\outer\source\database\  
> create-arg.[bat|sh] {dbuser} {dbpass} {dbname}
```

```
> cd \live-latex-demo\build\  
> python create.py {dbuser} {dbpass} {dbname}
```

Output: \live-latex-demo\generated\language.latin2.tex

```
> cd \live-latex-demo\  
> pdflatex live-latex-demo.tex
```

Output: \live-latex-demo\live-latex-demo.pdf

Irodalomjegyzék

- ELTE, Dokumentálás
(http://progalap.elte.hu/downloads/seged/eTananyag/lecke28_lap1.html)
Utolsó látogatás: 2017.03.12.
- BME LaTeX magyarosítás
(<http://math.bme.hu/latex/>)
Utolsó látogatás: 2017.04.03.